4. Regelbasierte Systeme

regelbasiertes Programmierparadigma:

Computer = Transformationsmaschine für Strukturen

Es gibt eine aktuelle Struktur, die solange transformiert wird, wie dies möglich ist.

Arbeitsprozess: Such- und Anwendungsprozess.
matching: Suchen einer passenden Regel,
rewriting: Anwendung der Regel, um die Struktur umzuschreiben.

Programm = Menge von Transformationsregeln

Programmfindung = Spezifikation der Regeln

Transformation von Strukturen:
- Aufbau komplexer Strukturen (z.B. Modelle für natürliche Objekte)
- Vereinfachung (z.B. Theorembeweis); *Reduktion* auf Normalform

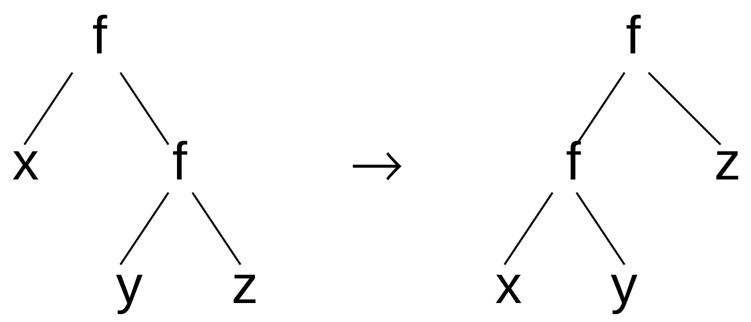
Beispiel für regelbasierte Systeme:

Termersetzungssysteme

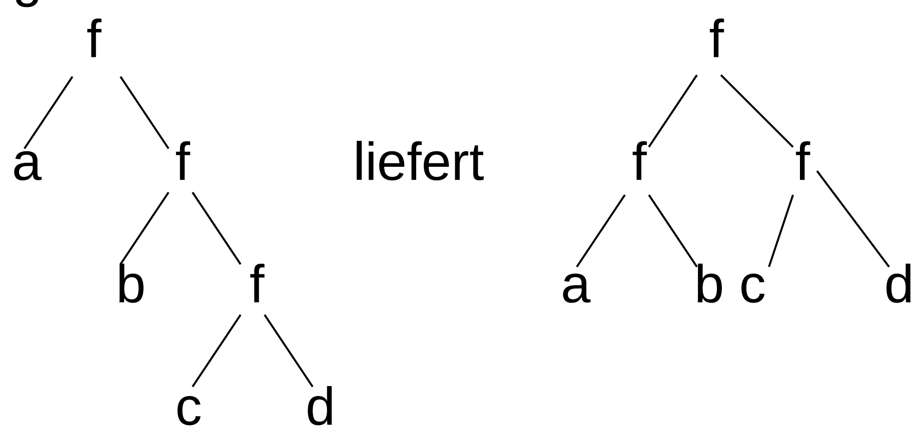
Sei Σ eine Signatur wie in der Prädikatenlogik (vgl. Kap. 1), $Term_\Sigma$ die Menge der Terme über Σ und $Ground_\Sigma$ die Menge der Grundterme (d.h. ohne Variablen) über Σ.

Ein *Termersetzungssystem* über Σ ist eine Teilmenge von $Term_\Sigma \times Term_\Sigma$. Die einzelnen Elemente (die *Regeln*) werden in der Form $T_1 \rightarrow T_2$ geschrieben. Die *Anwendung* einer solchen Regel (Reduktionsrelation) auf einen Grundterm besteht aus der Ersetzung eines Subterms, der bei passender Substitution σ mit T_1 matcht, durch $\sigma(T_2)$. Beispiel:

Regel



Anwendung auf



Spezialfall: ***Stringersetzungssysteme***
(auch: Semi-Thue-Systeme)

Die Grundterme sind hier die endlichen Wörter über einem Alphabet Σ.
Die Regeln sind von der Form $\alpha \to \beta$, wobei $\alpha, \beta \in \Sigma^*$.
Eine *Anwendung* einer Regel (Reduktionsschritt) besteht aus der Ersetzung eines Teilwortes α durch die zugehörige rechte Regelseite, β.
Beispiel:
$\Sigma = \{ A; B \}$, einzige Regel BA $\to$ AABB.
Mit Startwort BAA erhält man sukzessive:

BAA
AABBA
AABAABB
AAAABBABB
AAAABAABBBB
...

(matchende linke Regelseite jeweils rot markiert)

$\Rightarrow$ dieses Stringersetzungssystem ist offensichtlich nicht terminierend.

Anderes Beispiel:
$\Sigma = \{ A; B \}$, einzige Regel BBA $\to$ AAAB.

Dieses System ist terminierend, da die Anzahl der B bei jedem Reduktionsschritt abnimmt (und nicht kleiner als 0 werden kann).

Es ist bisher nicht bekannt, ob die Termination von Stringersetzungssystemen mit nur einer Regel generell entscheidbar ist.

Formale Grammatiken

Eine *Chomsky-Grammatik* (Typ-0-Grammatik) ist ein Stringersetzungssystem, für das im Alphabet eine Teilmenge *N* von Nichtterminalzeichen ausgezeichnet ist und bei der in jeder linken Regelseite mindestens 1 Nichtterminalzeichen auftritt.

Eine *Typ-1-Grammatik* (auch: *kontextsensitive* Grammatik) ist eine Typ-0-Grammatik, in der alle Regeln die Form $\alpha A\beta \rightarrow \alpha\gamma\beta$ (mit $A \in N$; $\alpha, \beta, \gamma \in \Sigma^*$) haben.

Eine *Typ-2-Grammatik* (auch: *kontextfreie* Grammatik) ist eine Typ-0-Grammatik, in der die linke Seite bei allen Regeln nur aus genau einem Nichtterminalzeichen besteht.

Eine *Typ-3-Grammatik* (auch: *reguläre* Grammatik) ist eine Typ-2-Grammatik, bei der auf der rechten Regelseite höchstens ein Terminalzeichen auftreten darf und – wenn dies der Fall ist – allenfalls noch *ein* Nichtterminalzeichen. Die erlaubte Stellung solcher Nichtterminalzeichen muss außerdem über alle Regeln hinweg einheitlich *immer vor* oder *immer hinter* dem Terminalzeichen sein.

Als Bestandteil einer formalen Grammatik zählt außerdem ein Startzeichen (auch *Axiom* genannt) aus der Menge *N* der Nichtterminalzeichen.

Die von einer Grammatik erzeugte Sprache ist die Menge aller Wörter, die aus dem Startzeichen durch sequenzielle Anwendung der Regeln in endlich vielen Schritten abgeleitet werden können und die nur aus Terminalzeichen bestehen.

Chomsky-Hierarchie:
Typ-0-Sprachen ⊃ kontextsensitive Sprachen ⊃ kontextfreie Sprachen ⊃ reguläre Sprachen.

(Quellenangaben siehe http://www.uni-forst.gwdg.de/~wkurth/fs10_lit.htm)